Neural networks, especially complex neural networks, are resource-hungary algorithms. When it comes to training neural networks on medium to large datasets, the amount of computation time to adequately train a model can take hours (or even days!) With simple modelling problems, like the ones covered in this module, training a model in the same notebook as an analysis is no problem. However, with more formal applications of neural network and deep learning models, data scientists cannot afford the time or resources to build and train a model each time they analyze data. In these cases, a trained model must be stored and accessed outside of the training environment.

With TensorFlow, we have the ability to save and load neural network models at any stage, including partially trained models. When building a TensorFlow model, if we use Keras' ModelCheckpoint method, we can save the model weights after it tests a set number of data points. Then, at any point, we can reload the checkpoint weights and resume model training. Saving checkpoints while training has a number of benefits:

* We can short-circuit our training loop at any time (stop the function by pressing CTRL+C, or by pressing the stop button at the top of the notebook). This can be helpful if the model is showing signs of overfitting.
* The model is protected from computer problems (power failure, computer crash, etc.). Worst-case scenario: We would lose five epochs' worth of optimization.
* We can restore previous model weight coefficients to try and revert overfitting.